UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/533,735	05/04/2005	Robert Jochemsen	NL 021103	4556

65913 7590 03/25/2009

NXP, B.V.
NXP INTELLECTUAL PROPERTY DEPARTMENT
M/S41-SJ
1109 MCKAY DRIVE
SAN JOSE, CA 95131

EXAMINER	
SCHNEE, HAL W	

ART UNIT	PAPER NUMBER
2186	

NOTIFICATION DATE	DELIVERY MODE
03/25/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ip.department.us@nxp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/533,735
Filing Date: May 04, 2005
Appellant(s): JOCHEMSEN ET AL.

Mark A. Wilson
Reg. No. 43,994
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7 January 2009 appealing from the Office action mailed 7 August 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2003/0182414	O'Neill	09-2003
6,804,763	Stockdale et al.	10-2004
5,701,516	Cheng et al.	12-1997
2003/0081932	Hanes	05-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-4, 7, 8, and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stockdale et al. (U.S. Patent 6,804,763, hereafter "Stockdale") in view of Cheng et al. (U.S. Patent 5,701,516, hereafter "Cheng"), and further in view of O'Neill (U.S. 2003/0182414).

Regarding Claim 1, Stockdale teaches a memory management device for managing a memory space of at least one persistent-memory device (fig. 4; col. 4, lines 15-17—"non-volatile memory storage device"), comprising a memory allocation unit (col. 10, lines 52-55—"NV-RAM manager") adapted to communicate with at least one application device and to allocate at least one first part of said memory space to said application device (col. 4, lines 17-25. Application devices are shown in fig. 2, item 215; the applications recited in col. 7, lines 19-34, such as the word processing program or the graphical utility program, may also be considered the application device), wherein said allocation unit is further adapted to communicate with at least one file system device, and to allocate on request from said application device or from said file system device said first part of said memory space to said file system (col. 7, lines 19-26 describes the file system. The operating system described in col. 10, lines 54-67 includes utilities for accessing and manipulating files; it may therefore be considered a file system device. Likewise, col. 11, lines 1-12 describes various clients, which are processes such as a virtual player tracking unit and a bank manager, and other software units. These processes and units also access data stored in the file system, and may be considered file system devices).

Stockdale does not teach that the persistent memory is used as a write cache memory for said file system. However, Cheng teaches a persistent memory used as a write cache memory for said file system (col. 2, lines 57-61).

All of these claimed elements were known in Stockdale and Cheng and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the persistent memory used as write cache of Cheng with the device of Stockdale to yield the predictable result of a memory management device in which the persistent memory is used as a write cache for a file system device. One would be motivated to make this combination for the purpose of increasing the speed of writes by writing data to be stored to the non-volatile memory rather than a slower storage device.

Stockdale/Cheng does not specifically teach using the allocated memory space to write a first working data structure comprising a plurality of working data blocks to the memory space and to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure.

However, O'Neill teaches using the allocated memory space to write a first working data structure comprising a plurality of working data blocks to the memory space and to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure (fig. 10, state 1260, memory blocks 1234 and 1242; ¶ [0148]—first working data structure 1234 is written to persistent {flash}

memory 1002. An additional copy is made in the form of a second working data structure to location 1242 of persistent memory 1002 to provide additional fault tolerance. Since 1234 and 1242 are contained in the same persistent memory 1002, they can be considered as residing in the same memory space; since they are identical copies of the same data, the second working data structure comprises a copy of the working data blocks of the first working data structure).

All of the claimed elements were thus known in Stockdale/Cheng and O'Neill and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the working data structures of O'Neill with the persistent memory allocation of Stockdale/Cheng to yield the predictable result of a memory management device that allocates a part of the persistent memory to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure. One would be motivated to make this combination for the purpose of securing a copy of the working data while continuing to work on the original working data.

Regarding Claim 2, Stockdale teaches said memory allocation unit is adapted to maintain a memory allocation table at a current status, said memory allocation table assigning at least one memory address representing a defined part of said memory space to either said application device or to said file system device (fig. 9; col. 29, lines 17-19).

Regarding Claim 3, Stockdale teaches a memory management device according to claim 2, further comprising a processor and a memory (fig. 3, processor 300 and memories 325 and

355), wherein said memory allocation unit is implemented in the form of at least one first executable file contained in said memory (col. 10, lines 52-55—"NV-RAM Manager").

Regarding Claim 4, Stockdale teaches said memory is a persistent-memory device, in particular said persistent-memory device (col. 33, lines 11-14—software is installed in the NV-RAM and executed on the processor, as shown in fig. 11, item 1140).

Regarding Claim 7, Stockdale teaches an application device, comprising a persistent-memory device connected to a processor (fig. 3, processor 300 and persistent-memory device 355; col. 17, line 64-col. 18, line 3. The applications recited in col. 7, lines 19-34, such as the word processing program or the graphical utility program, may also be considered the application device), and a data management unit adapted to manipulate data in said persistent memory device (fig. 4; col. 4, lines 15-17—"non-volatile memory storage device"), wherein said data management unit is adapted to write at least one third executable file to said persistent memory device, or to provide the file system with a reference to at least on third executable file in said file system (col. 17, lines 38-41 shows adding an executable file to the persistent memory device; col. 35, lines 21-34 shows that the device can add any number of executable files), such that by executing said third executable file said processor is adapted to transform said data into a predetermined data-sequence form (col. 7, lines 45-49 and col. 35, lines 14-20—the processor can execute a compression utility, which transforms data into a predetermined data-sequence form; all executable files used by the processor are stored in the persistent memory device).

Stockdale does not teach that the persistent memory is used as a write cache memory for said file system. However, Cheng teaches a persistent memory used as a write cache memory for said file system (col. 2, lines 57-61).

All of these claimed elements were known in Stockdale and Cheng and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the persistent memory used as write cache of Cheng with the device of Stockdale to yield the predictable result of an application device in which the persistent memory is used as a write cache for a file system device. One would be motivated to make this combination for the purpose of increasing the speed of writes by writing data to be stored to the non-volatile memory rather than a slower storage device.

Stockdale/Cheng does not teach that the copy of the first working data structure is stored in a same memory space as the first working data structure. However, O'Neill teaches making a copy of the first working data structure is stored in a same memory space as the first working data structure (fig. 10, state 1260, memory blocks 1234 and 1242; ¶ [0148]—first working data structure 1234 is written to persistent {flash} memory 1002. An additional copy is made in the form of a second working data structure to location 1242 of persistent memory 1002 to provide additional fault tolerance. Since 1234 and 1242 are contained in the same persistent memory 1002, they can be considered as residing in the same memory space; since they are identical copies of the same data, the second working data structure comprises a copy of the working data blocks of the first working data structure).

All of the claimed elements were thus known in Stockdale/Cheng and O'Neill and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the working data structures of O'Neill with the data form transformation of

Stockdale/Cheng to yield the predictable result of an application device in which the processor is adapted to transform a copy of a first working data structure into a predetermined data-sequence form, wherein the copy of the first working data structure is stored in a same memory space as the first working data structure. One would be motivated to make this combination for the purpose of securing a copy of the working data while continuing to work on the original working data.

Regarding Claim 8, Stockdale teaches said data management unit is provided in the form of least one fourth executable file in a memory, particularly, in said persistent memory (col. 10, lines 52-55—"NV-RAM Manager" and col. 33, lines 11-14—software is installed in the NV-RAM and executed on the processor, as shown in fig. 11, item 1140).

Regarding Claim 10, Stockdale teaches a data processing system (fig. 3), comprising a memory management device for managing a memory space of at least one persistent-memory device (fig. 4; col. 4, lines 15-17—"non-volatile memory storage device"), comprising a memory allocation unit (col. 10, lines 52-55—"NV-RAM manager") adapted to communicate with at least one application device and to allocate at least one first part of said memory space to said application device (col. 4, lines 17-25. Application devices are shown in fig. 2, item 215; the applications recited in col. 7, lines 19-34, such as the word processing program or the graphical utility program, may also be considered the application device), wherein said allocation unit is further adapted to communicate with at least one file system device, and to allocate on request from said application device or from said file system device said first part of said memory space to said file system (col. 7, lines 19-26 describes the file system. The operating system described in col. 10, lines 54-67 includes utilities for accessing and manipulating files; it may therefore be

considered a file system device. Likewise, col. 11, lines 1-12 describes various clients, which are processes such as a virtual player tracking unit and a bank manager, and other software units. These processes and units also access data stored in the file system, and may be considered file system devices).

Stockdale does not teach that the persistent memory is used as a write cache memory for said file system. However, Cheng teaches a persistent memory used as a write cache memory for said file system (col. 2, lines 57-61).

All of the claimed elements were known in Stockdale and Cheng and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the persistent memory used as write cache of Cheng with the system of Stockdale to yield the predictable result of a data processing system in which the persistent memory is used as a write cache for a file system device. One would be motivated to make this combination for the purpose of increasing the speed of writes by writing data to be stored to the non-volatile memory rather than a slower storage device.

Stockdale/Cheng does not specifically teach using the allocated memory space to write a first working data structure comprising a plurality of working data blocks to the memory space and to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure.

However, O’Neill teaches using the allocated memory space to write a first working data structure comprising a plurality of working data blocks to the memory space and to write a

second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure (fig. 10, state 1260, memory blocks 1234 and 1242; ¶ [0148]—first working data structure 1234 is written to persistent {flash} memory 1002. An additional copy is made in the form of a second working data structure to location 1242 of persistent memory 1002 to provide additional fault tolerance. Since 1234 and 1242 are contained in the same persistent memory 1002, they can be considered as residing in the same memory space; since they are identical copies of the same data, the second working data structure comprises a copy of the working data blocks of the first working data structure).

All of the claimed elements were thus known in Stockdale/Cheng and O'Neill and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the working data structures of O'Neill with the persistent memory allocation of Stockdale/Cheng to yield the predictable result of a data processing system that allocates a part of the persistent memory to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure. One would be motivated to make this combination for the purpose of securing a copy of the working data while continuing to work on the original working data.

Claims 5, 6, and 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stockdale (U.S. Patent 6,804,763) in view of Cheng (U.S. Patent 5,701,516), further in view of O'Neill (U.S. 2003/0182414) and further in view of Hanes (U.S. 2003/0081932).

Regarding Claim 5, Stockdale teaches a file system device adapted to maintain a file allocation table at a current status, said file allocation table assigning at least one disk space address to at least one file, wherein said file allocation Unit is adapted to communicate with a memory management device that is related to a persistent-memory device and to include an address of at least one first memory space of said persistent-memory device in the maintenance of said file allocation table (col. 34, lines 51-58—the same techniques are used for the file system device as for memory management for the application device. Fig. 12 and col. 35, lines 14-20 show the files; fig. 9 shows the file allocation table).

In the device of Stockdale, the functions of the file system device are performed by the same processor as the functions of the application device. Stockdale thus does not specifically teach a separate file system device, comprising a file allocation unit; or that the persistent memory is used as a write cache memory for said file system device. However, Cheng teaches a file system device, comprising a file allocation unit (fig. 2, Storage Processor 38 {shown in detail in fig. 3}; col. 5, lines 34-37); and the persistent memory is used as a write cache memory for said file system device (col. 2, lines 57-61).

All of these claimed elements were known in Stockdale and Cheng and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the file system device comprising a file allocation unit and the persistent memory used as write

cache of Cheng with the device of Stockdale to yield the predictable result of a file system device with a file allocation unit, in which the persistent memory is used as a write cache for the file system device. One would be motivated to make this combination for the purpose of increasing the speed of writes by writing data to be stored to the non-volatile memory rather than a slower storage device, and to free the application device to perform other tasks by offloading file management to a file allocation unit.

Stockdale/Cheng teaches converting a copy of working data into a predetermined data-sequence form (Stockdale, col. 7, lines 45-49 and col. 35, lines 14-20—the processor can execute a compression utility, which transforms data into a predetermined data-sequence form; all executable files used by the processor are stored in the persistent memory device), but does not specifically teach wherein the file system device is configured to convert a copy of a first working data structure to a file data structure and to write the file data structure to a secondary storage medium, wherein the copy of the first working data structure is written to a same memory space as the first working data structure.

However, O'Neill teaches using the allocated memory space to write a first working data structure comprising a plurality of working data blocks to the memory space and to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure (fig. 10, state 1260, memory blocks 1234 and 1242; ¶ [0148]—first working data structure 1234 is written to persistent {flash} memory 1002. An additional copy is made in the form of a second working data structure to location 1242 of persistent memory 1002 to provide additional fault tolerance. Since 1234 and

1242 are contained in the same persistent memory 1002, they can be considered as residing in the same memory space; since they are identical copies of the same data, the second working data structure comprises a copy of the working data blocks of the first working data structure).

All of these claimed elements were known in Stockdale/Cheng and O'Neill and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the working data structures of O'Neill with the persistent memory allocation of Stockdale/Cheng to yield the predictable result of a file system device that allocates a part of the persistent memory to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure. One would be motivated to make this combination for the purpose of securing a copy of the working data while continuing to work on the original working data.

Stockdale/Cheng/O'Neill does not specifically teach the file system device is configured to convert a copy of a first working data structure to a file data structure and to write the file data structure to a secondary storage medium. However, Hanes teaches the file system device is configured to convert a working data structure to a file data structure and to write the file data structure to a secondary storage medium (fig. 2A; ¶ [0017]—CD Recorder Engine 20 {file system device} converts working data from application formats to a file system format for storage on a CD. An embodiment of the interface between the application device and the file system device is described in ¶ [0014], and recording to CD {writing the file structure to a secondary storage medium} is described in ¶ [0019]).

All of the claimed elements were thus known in Stockdale/Cheng/O'Neill and Hanes and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the conversion to a file system format and writing to CD of Hanes with the working data copy of Stockdale/Cheng/O'Neill to yield the predictable result of a file system device that is configured to convert a copy of a first working data structure to a file data structure and to write the file data structure to a secondary storage medium. One would be motivated to make this combination for the purpose of securing the working data on a secondary medium for long-term storage.

Regarding Claim 6, Stockdale teaches a file system device according to claim 5, further comprising a processor and a memory (fig. 3, processor 300 and memories 325 and 355), wherein said memory allocation unit is implemented in the form of at least one second executable file contained in said memory (col. 10, lines 52-55—"NV-RAM Manager").

Regarding Claim 11, Stockdale teaches a method for managing memory space of a persistent-memory device (col. 6, lines 5-6), comprising allocating at least one first part of said memory space to a file system device (col. 6, lines 11-13 and col. 7, lines 23-26); and

converting a working data structure into a predetermined data sequence form (col. 7, lines 45-49 and col. 35, lines 14-20—the processor can execute a compression utility, which converts data into a predetermined data-sequence form).

Stockdale does not teach that the persistent memory is used as a write cache memory for said file system. However, Cheng teaches a persistent memory used as a write cache memory for said file system (col. 2, lines 57-61).

All of these claimed elements were known in Stockdale and Cheng and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the persistent memory used as write cache of Cheng with the system of Stockdale to yield the predictable result of a data processing system in which the persistent memory is used as a write cache for a file system device. One would be motivated to make this combination for the purpose of increasing the speed of writes by writing data to be stored to the non-volatile memory rather than a slower storage device.

Stockdale/Cheng does not teach:

writing a first working data structure comprising a plurality of working data blocks to the memory space; and

writing a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure.

However, O'Neill teaches:

writing a first working data structure comprising a plurality of working data blocks to the memory space; and writing a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure (fig. 10, state 1260, memory blocks 1234 and 1242; ¶ [0148]—first working data structure 1234 is written to persistent {flash} memory 1002. An additional copy is made in the form of a second working data structure to location 1242 of persistent memory 1002 to provide additional fault tolerance.

Since 1234 and 1242 are contained in the same persistent memory 1002, they can be considered as residing in the same memory space; since they are identical copies of the same data, the second working data structure comprises a copy of the working data blocks of the first working data structure).

All of the claimed elements were thus known in Stockdale/Cheng and O'Neill and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the working data structures of O'Neill with the memory space and converting a working data structure into a predetermined data sequence form of Stockdale/Cheng to yield the predictable result of a method for managing memory space of a persistent memory device that allocates a part of the persistent memory to write a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure, and converts the second working data structure into a predetermined data sequence form. One would be motivated to make this combination for the purpose of securing a copy of the working data while continuing to work on the original working data.

Regarding Claim 12, Stockdale teaches said allocating step comprises a step of blocking a writing access to said first part of said memory space (col. 6, lines 54-61 and col. 32, lines 22-24—restricting access means blocking writing access for certain devices or processes).

Regarding Claim 13, Stockdale teaches said allocating step comprises a step of giving away to said file system device the power of reading access to said first part of said memory

space (col. 6, lines 46-50—sending a handle constitutes giving away reading access since the handle is required to access the allocated memory area).

Regarding Claim 14, Stockdale teaches a step of deallocating said first part of said memory space to a memory management device (col. 6, lines 41-43 and 54-58—the method includes deallocating among the functions).

Regarding Claim 15, Stockdale teaches said allocating step or said deallocating step comprises transmitting an address range defining said first part of said memory space from said memory management device to said file system device or, respectively, vice versa (col. 28, lines 18-20 and col. 29, lines 22-24—the method transmits a handle to a memory block of predefined size; this defines the range of addresses that is transmitted).

Claims 16-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stockdale (U.S. Patent 6,804,763) in view of Cheng (U.S. Patent 5,701,516) in view of O’Neill (U.S. 2003/0182414) in view of Hanes (U.S. 2003/0081932), as applied to Claim 14, above, and further in view of Lee et al. (U.S. Patent 5,930,167, hereafter “Lee”).

Regarding Claim 16, Stockdale/Cheng/O’Neill/Hanes teaches said deallocating step is performed for said first part of said memory space based on any number of conditions (Stockdale, col. 32, lines 60-65—various flags govern when deallocation is and is not allowed).

Stockdale/Cheng/O’Neill/Hanes does not specifically teach that deallocation is performed given the condition that first data contained in said first part of said memory space is stored in the form of file data in a second part of said memory space, said file data having a predetermined

file structure, and that said second part of said memory space is allocated to said file system device.

However, Lee teaches that first data contained in said first part of said memory space is stored in the form of file data in a second part of said memory space, said file data having a predetermined file structure, and that said second part of said memory space is allocated to said file system device (col. 5, line 60-col. 6, line 6; figs. 4A-4D).

All of the claimed elements were known in Stockdale/Cheng/O'Neill/Hanes and Lee, and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the storing data of Lee with the conditional deallocation of Stockdale//Cheng/O'Neill/Hanes to yield the predictable result of performing the deallocating step given the condition that first data contained in said first part of said memory space is stored in the form of file data in a second part of said memory space, said file data having a predetermined file structure, and that said second part of said memory space is allocated to said file system device. One would be motivated to make this combination for the purpose of securing a copy of the working data before deallocating the memory used to store the working data.

Regarding Claim 17, Stockdale/Cheng/O'Neill/Hanes teaches said deallocating step is performed for said second part of said memory space given the condition that said file data has been written to a secondary storage medium (Cheng, col. 6, lines 42-48—the flush operation writes the data to a disk, a secondary storage medium).

Regarding Claim 18, Stockdale/Cheng/O'Neill/Hanes teaches the step of Claim 17, as described above, but does not teach a method for write-caching first data worked on by an

application, said first data being contained in a first part of a memory space of a persistent-memory device. However, Lee teaches a method for write-caching first data worked on by an application, said first data being contained in a first part of a memory space of a persistent-memory device (Lee, Abstract, lines 1-6 and col. 8, lines 10-14).

All of the claimed elements were known in Stockdale/Cheng/O’Neill/Hanes and Lee, and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the method for write caching of Lee with the method of Stockdale/Cheng to yield the predictable result of performing a memory managing method according to claim 17comprised in a method for write-caching first data worked on by an application, said first data being contained in a first part of a memory space of a persistent-memory device.

Regarding Claim 19, Stockdale/Cheng/O’Neill/Hanes teaches after said allocating step, a step of sending a confirmation message from said file system device to said application device (Cheng, col. 7, lines 50-59—after performing an operation, a confirmation message is sent to the application device {host processor}; and col. 13, lines 51-55—enabling the NVRAM constitutes an allocating step).

Regarding Claim 20, Stockdale/Cheng/O’Neill/Hanes teaches flags determining when deallocation is allowed (Stockdale, col. 32, lines 60-65) and additional data stored in the persistent memory (col. 14, lines 44-50 and 62-63; this constitutes the third data of the present claim), but Stockdale/Cheng/O’Neill/Hanes does not teach said first data is a copy of third data contained in a third part of said memory space, said write-caching method comprising, before

performing said memory managing method, a step of copying said third data to said first memory space.

However, Lee teaches said first data is a copy of third data contained in a third part of said memory space, said write-caching method comprising a step of copying said third data to said first memory space (col. 5, line 60-col. 6, line 6; figs. 4A-4D shows the copying operation before the memory management method {i.e. before deallocating the memory}).

All of the claimed elements were known in Stockdale, Cheng, and Lee, and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the data copying and write caching method of Lee with the determination of when deallocation can be performed of Stockdale/Cheng to yield the predictable result of a write caching method that copies third data to the first memory space before performing a memory management method.

Regarding Claim 21, Stockdale/Cheng/O'Neill/Hanes teaches:

allocating a fourth part of said memory space to said application device for an executable file or dynamic link library (Stockdale, col. 17, lines 38-41 shows adding an executable file to the persistent memory device; col. 35, lines 21-34 shows that the device can add any number of executable files);

writing said executable file or dynamic link library to said fourth part of said memory space (Stockdale, col. 33, lines 11-14—any number of executables can be stored in any number of parts of the memory space); and

allocating said fourth part of said memory space to said file system device (Stockdale, col. 7, lines 19-26 and col. 34, lines 51-58—any number of parts of the memory space can be allocated to the file system device).

Stockdale/Cheng/O’Neill/Hanes does not teach that the executable file is adapted to converting said first data into file data. However, Lee teaches the executable file is adapted to converting said first data into file data (col. 5, line 60-col. 6, line 6 and figs. 4A-4D show an executable adapted to converting data into file data).

All of the claimed elements were known in Stockdale/Cheng/O’Neill/Hanes and Lee and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the executable file of Lee with the memory allocation and writing of Stockdale/Cheng to yield the predictable result of method that allocates memory to the file system device and writes an executable file that is adapted to converting data into file data.

Regarding Claim 22, Stockdale/Cheng/O’Neill/Hanes does not teach a step of transforming said first data into said file data with the aid of said executable file or said dynamic link library. However, Lee teaches a step of transforming said first data into said file data with the aid of said executable file or said dynamic link library (col. 5, line 60-col. 6, line 6).

All of the claimed elements were known in Stockdale/Cheng/O’Neill/Hanes and Lee and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the transforming data of Lee with the method of Stockdale/Cheng/O’Neill/Hanes to

yield the predictable result of method that includes a step of transforming said first data into said file data with the aid of said executable file or said dynamic link library.

Regarding Claim 23, Stockdale/Cheng/O'Neill/Hanes does not teach said transforming step is initiated by said file system device. However, Lee teaches said transforming step is initiated by said file system device (col. 6, lines 12-19—performing the transforming step when the memory system is idle is necessarily initiated by the file system device since the application device has no way of knowing when the memory system is idle).

All of the claimed elements were known in Stockdale/Cheng/O'Neill/Hanes and Lee and could have been combined by known methods with no change in their respective functions. It therefore would have been obvious to a person of ordinary skill in the art at the time of invention to combine the initiation of Lee with the method of Stockdale/Cheng to yield the predictable result of method in which the transforming step is initiated by the file system device.

Regarding Claim 24, Stockdale/Cheng/O'Neill/Hanes teaches after said transforming step, a step of deallocating said fourth part of said memory space to said memory management device (Stockdale, col. 6, lines 41-43 and 54-58—the method includes deallocating among the functions, and it can be performed after any step in the process).

Regarding Claim 25, Stockdale/Cheng/O'Neill/Hanes teaches a method for saving data worked on by an application device to a file on a secondary storage medium (Cheng, col. 2, lines 57-61—data storage devices are a secondary storage medium), comprising writing said file data to said secondary storage medium (Cheng, col. 7, lines 25-29).

(10) Response to Argument

On pages 8-9 of the Appeal Brief, the appellant argues that O'Neill does not teach writing a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same persistent memory device (or memory space) as the first working data structure, as recited by Claim 1.

The examiner first notes that the version of the present claim under appeal properly recites " . . . in the same memory space . . .," as contained in the arguments of page 9 of the Appeal Brief. The recitation of " . . . in the same persistent memory device . . ." appears in the amendment filed 7 October 2008; the examiner did not enter this amendment because it changed the scope of the claimed invention. The examiner believes that the inclusion of "the same persistent memory device" on page 8 was a typographical error.

Paragraphs [0139] - [0152] of O'Neill describe a series of operations performed on data, shown pictorially in figure 10. As the appellant states, many of these operations are performed based on working data structure 1232, which is stored in RAM 1004. The appellant therefore argues that a copy of this data structure stored in Flash memory 1002 does not meet the limitations of the present claims because the copy is not written to the same memory device, and hence not the same memory space.

However, in formulating the rejections above, the examiner has relied specifically on state 1260 in figure 10, described in paragraph [0148]. As explained therein, a copy of the working data structure 1232 is written to Flash memory 1002; this copy is called 1234. It is data structure 1234 that the examiner has equated to the "first working data structure" of the present

claims. This meets the requirements of the limitation "to write a first working data structure comprising a plurality of working data blocks to the memory space." 1234 may be considered a working data structure because, as described in paragraphs [0148] and following, further operations may be performed on the data in 1234.

As further explained in paragraph [0148], a second copy of data structure 1232 is then written to Flash memory 1004; this copy is called 1242. Since both 1234 and 1242 are duplicates of the same data structure, 1242 is an exact copy of 1234. Both data structures 1234 and 1242 are contained in the same memory space (and the same persistent-memory device) in Flash memory 1002. Data structure 1242 therefore constitutes "a second working data structure comprising a copy of the plurality of working data blocks, wherein the second working data structure comprises a copy of the first working data structure in the same memory space as the first working data structure," as recited by the present claim.

On page 10 of the Appeal Brief, the appellant argues that Stockdale does not teach an allocation unit that is adapted to communicate with at least one file system device. The appellant asserts that Stockdale does not show a file system device that is separate from the application device, and that the file system device must have a device driver which is not shown by Stockdale. With respect to these arguments, it is the examiner's perspective that the appellant is viewing the present claim too narrowly and is arguing for features that are not claimed.

On the first page of the appellant's specification, lines 19-27 explain that a file system device is not necessarily a hardware device. To the contrary, "devices" are clearly described as being software entities. As the relevant portion of the cited passage states:

> "The term application as used herein refers to a process or a device adapted to manipulate memory data . . . An application is implemented as a device in the form of one or a number of executable

> files loaded into a memory that is connected to a central processing unit . . . Note that a file system and a memory manager are to be considered special cases of an application."

So the file system device, like the application device and memory manager, may be software programs. It is with this understanding that Stockdale teaches the limitations in question. Stockdale teaches several such applications that may be considered to be the file system device of the present claim. The operating system described in col. 10, lines 54-67 includes utilities for accessing and manipulating files; it may therefore be considered a file system device. Likewise, col. 11, lines 1-12 describes various clients, which are processes such as a virtual player tracking unit and a bank manager, and other software units. These processes and units also access data stored in the file system, and may be considered file system devices. Others of these processes or units may be considered the application device of the present claim; the applications recited in col. 7, lines 19-34, such as the word processing program or the graphical utility program, may also be considered the application device.

As explained in col. 11, lines 1-12, in the described embodiment, all processes access the file system through the NV-RAM manager (the memory allocation unit of the present claim). Stockdale therefore teaches said allocation unit is adapted to communicate with at least one file system device, and to allocate on request from said application device or said file system device said first part of said memory space to said file system, as recited by the present claim.

As for the appellant's argument that Stockdale does not show a device driver for the file system device, the present claim does not require a device driver, and Stockdale shows that none is required to perform the operations recited by the claim. The appellant also argues that Stockdale merely stores working data in non-volatile memory, but does not store data on a file system device. The present claim, however, does not recite storing data on the file system device;

it merely requires that the file system device be in communication with the memory allocation device and make requests for memory allocation. (Given that the file system device is a software entity, as described in the appellant's specification, storing data on the file system device would not make much sense.) Standard file system commands, such as storing a file, as described in col. 7, lines 19-26, constitute a request for memory allocation in the file system. Stockdale also provides additional detail regarding memory allocation in col. 28, lines 18-37.

The examiner therefore maintains that the combination of Stockdale, Cheng, and O'Neill teaches all of the limitations of Claim 1. On page 11 of the Appeal Brief, the appellant argues that the prior art of record does not teach all of the limitations of Claim 7 because it includes a similar limitation regarding writing a second working data structure in the same memory space as the first data structure. The examiner maintains that Stockdale teaches this limitation in the same manner as described for Claim 1, above; the prior art of record therefore teaches all of the limitations of Claim 7. Similar reasoning also applies to Claim 10. Since the examiner maintains the rejections of independent Claims 1 and 7, the rejections of Claims 2-4 and 8, which depend upon 1 and 7, respectively, are also maintained.

On page 12 of the Appeal Brief, the appellant argues that the combination of Stockdale, Cheng, O'Neill, and Hanes does not teach all of the limitations of Claims 5, 6, and 11-15. The following paragraphs of the Appeal Brief detail the ways in which the appellant believes the prior art of record does not teach the limitations of Claim 5. On page 13, the appellant argues that O'Neill does not teach converting a copy of a first working data structure to a file data structure and to write the file data structure to a secondary storage medium. However, the examiner has not relied upon O'Neill to teach these limitations of Claim 5. As seen on page 12 of the Final

Office Action mailed 7 August 2008, and shown in the rejections above, the examiner has relied upon paragraphs [0014], [0017], and [0019] of Hanes for these teachings. In paragraph [0017], Hanes clearly teaches a CD Recorder Engine 20 (a file system device) that converts working data from application formats to a file system format for storage on a CD; paragraph [0019] shows recording to the CD, which writes the file structure to a secondary storage medium.

The examiner notes that these limitations in Claim 5 are more restrictive than limitations in Claim 7 and 11 that appear to be similar. In particular, the latter claims do not include writing to a secondary storage medium. In addition, they do not specify that the working data structure is converted into a file data structure; in contrast, they recite converting or transforming the working data structure into "a predetermined data-sequence form." This is a much broader limitation than that recited by Claim 5. In Claim 11, for example, the examiner has relied upon Stockdale to teach converting a working data structure into a predetermined data sequence form. In column 7, lines 45-49 and col. 35, lines 14-20, Stockdale explains that the processor can execute a compression utility to compress data in the non-volatile memory. The compression process converts data into a predetermined data-sequence form.

The examiner further notes that O'Neill would also be able to teach these limitations in Claims 7 and 11 because a predetermined data-sequence form may be *any* data-sequence form, as long as that form is known in advance. As explained in paragraph [0147], O'Neill applies instructions to update the information contained in a working data structure. An overview of this procedure is described in paragraphs [0136] and [0137]; as explained, code is executed which modifies (i.e. converts or transforms) the information in the working data structure to produce a pre-determined result. Since the working data structure is copied to a second working data

structure, as described in reference to Claim 1, above, O'Neill is converting/transforming the second working data structure into a predetermined data-sequence form.

Returning to the arguments for Claim 5, on pages 13-14 of the Appeal Brief, the appellant argues that Stockdale does not teach assigning at least one disk space address to at least one file, asserting that Stockdale does not teach any form of a disk storage device. The examiner agrees that Stockdale does not specifically show a disk device attached to the described gaming system. However, column 34, lines 51-68 clearly describes the implementation of a system involving disk storage devices. As explained, the non-volatile memory file system used by Stockdale is substantially similar to file systems on a standard computer hard drive, floppy drive, or CD-ROM drive; and the file system is organized in the same manner. The following paragraphs (column 35, lines 1-44) further detail the file system operations, stating that the blocks of non-volatile memory, which appear as files in the non-volatile memory system, can be copied, removed, renamed, or resized just as any file on a hard drive (column 35, lines 12-14.) A person of ordinary skill in the art would therefore recognize that assigning an address to a file in the non-volatile memory system of Stockdale—which would use standard file system operations—is equivalent to assigning a disk space address to a file, as recited by the present claim. In this way, Stockdale teaches assigning at least one disk space address to at least one file.

On page 14 of the Appeal Brief, the appellant argues that Stockdale does not teach the copy of the first working data structure is written to a same memory address space as the first working data structure. The examiner maintains that Stockdale teaches this limitation in Claim 5 in a similar manner as for Claim 1, detailed above.

On pages 14-15 of the Appeal Brief, the appellant argues that the combination of Stockdale, Cheng, O’Neill, and Hanes does not teach all of the limitations of Claim 11, in a similar manner as for Claim 5. The examiner’s reasoning for Claim 5, above, applies to Claim 11; and, as explained above, the limitations of Claim 11 related to converting data into a predetermined data-sequence form are broader than those of Claim 5, allowing either Stockdale (as used in the rejections above) or O’Neill to teach these limitations in Claim 11. Since the rejections of Claim 5 and 11 are maintained, the examiner is also maintaining the rejections of Claims 6 and 12-15, which depend upon Claims 5 and 11, respectively.

On page 15 of the Appeal Brief, the appellant argues that Claims 16-25 are patentable over Stockdale, Cheng, O'Neill, and Hanes in a similar manner to Claim 11. Since the rejection of Claim 11 is maintained, the examiner is also maintaining the rejections of Claims 16-25, which depend upon Claim 11.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner’s answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Hal Schnee

Conferees:

Kevin Ellis

/Kevin L Ellis/
Acting SPE of Art Unit 2187

/Matt Kim/
Supervisory Patent Examiner, Art Unit 2186